

Struts2 OGNL 表达式语言的研究与应用

聂常红¹, 唐远强², 冯相忠³

- (1. 广州大学华软软件学院, 广东 广州 510990;
2. 浙江海洋学院 数理与信息学院, 浙江 舟山 316000;
3. 浙江海洋学院 机电工程学院, 浙江 舟山 316000)

摘 要:OGNL(Object-Graph Navigation Language)是 Struts2 默认的表情式语言,是 Struts2 的核心内容之一。为了让读者快速掌握进而熟练使用 Struts2 OGNL,文章首先深刻剖析了 OGNL 的工作原理,接着讨论了 OGNL 的使用场合以及常见的几种应用,最后重点以一个电子商务系统中的产品发布管理模块的实现为例说明 OGNL 的使用、数据类型转换以及数据在 View 层和 Controller 层之间的流转处理,为使用 Struts2 开发信息管理系统以及对 Struts2 的教学提供了较高的理论指导和实践参考价值。实践证明,使用 OGNL 可以极大地提高 Struts2 对 Web 应用的表现能力。

关键词:Struts2;OGNL;框架;MVC 模式;valueStack

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2013)03-0183-04

doi:10.3969/j.issn.1673-629X.2013.03.046

Research and Application of Struts2 OGNL Expression Language

NIE Chang-hong¹, TANG Yuan-qiang², FENG Xiang-zhong³

- (1. South China Institute of Software Engineering, Guangzhou 510990, China;
2. School of Mathematics, Physics & Information Science, Zhejiang Ocean University, Zhoushan 316000, China;
3. College of Mechanical and Electrical Engineering, Zhejiang Ocean University, Zhoushan 316000, China)

Abstract:OGNL is the Struts2 default expression language and one of Struts2 core content. In order to let reader quickly master Struts2 OGNL and skilfully use it, firstly analyzed the OGNL principle of work profoundly, then discussed when and how to use it; finally taked one product release management module for electronic commerce system as example, demonstrated how to use OGNL in detail, it is a high theoretical guidance and practical reference value for the development of management information system with Struts2 and the teaching Struts2. As a result, using OGNL can significantly enhance the Struts2 Web application presentation ability.

Key words:Struts2;OGNL;framework;MVC pattern;valueStack

0 引言

Struts2 是一个基于 MVC 模式的 Java Web 开发框架^[1~3],它提供了 MVC 中的视图和控制内容,主要用于实现 Web 应用体系结构中的表现层^[4]。为了简化表现层的开发,Struts2 框架使用了 Struts2 标签和 OGNL(Object-Graph Navigation Language)两种技术^[5,6]。OGNL 是 Struts2 的默认表达式语言,具有强大的功能,解决了 MVC 的核心内容所涉及的各个方面的问题。所以要使用好 Struts2 就必须熟练地掌握 OGNL。然而在实际的应用中,很多使用者尤其是初学者,往往会对 OGNL 的很多用法感到困惑不解,主

要原因有二点:一是对 OGNL 的工作原理不是很理解,再有就是没有熟练掌握 OGNL 的使用方法。为此文中将首先详尽剖析 OGNL 的工作原理并讨论 OGNL 常见的一些应用,最后将通过电子商务系统中的产品发布管理模块为例说明 OGNL 的使用。

1 Struts2 OGNL 的工作原理

Struts2 是目前应用最广泛的 MVC 框架的实现产品^[7]。在 MVC 框架中,数据是最为核心的内容,通过数据在 MVC 各个层次的流转,从而将 MVC 框架的不同层次联系起来。在数据的流转过程中,不同层次的数据类型有可能是不一致的,比如,View 层所有数据的类型都是字符串类型,Controller 层数据则是一个个强类型的 JAVA 对象。为解决数据在 View 层和 Controller 层之间流转时所涉及到的类型不匹配问题,Struts2 采纳了 XWork 的 OGNL 方案,并且在传统

收稿日期:2012-06-26;修回日期:2012-09-30

基金项目:浙江省科技计划资助项目(2008C31009)

作者简介:聂常红(1972-),女,广东韶关人,硕士,讲师,研究方向为软件工程、现代 Web 应用开发。

OGNL 的基础上,引入了 OGNL ValueStack(值栈)的机制,从而比较完美地解决了数据流转中的不匹配性^[8]。通过在 View 层中结合使用 Struts2 标签和 OGNL 表达式可完美地展现来自于控制层及模型层的数据。

ValueStack 是一个堆栈结构,其中包含了一组对象,对这组对象的操作按先入后出的顺序进行^[9,10]。Struts2 提供了一个特殊的 OGNL PropertyAccessor^[11],可实现对 ValueStack 从上往下的自动搜索,并返回第一个符合条件的对象元素。Struts2 OGNL 的根对象是 ValueStack,ValueStack 中所包含的所有对象都属于根对象^[9]。在 Struts2 中,action 实例都存储在一个 ValueStack 的区域里。

在 Struts2 中,对于每个请求,都建立一个与相应 Action 对应的 ActionContext 和 ValueStack^[12],其中 ActionContext 将作为 Struts2 OGNL 的上下文环境,它维护着一个 Map 的结构。在这个 Map 结构中,除了 ValueStack 根对象以外,Struts2 还在这个上下文环境中包括了被 XWork 封装成 Map 对象的 Servlet 对象:request、session、application、parameters 和 attr 等称为命名对象的非根对象。此外,在 Struts2 中,某些标签如 iterator、bean 等标签在使用了 var 属性后,会将新生成的对象放入 Stack Context 中,所以,在 ActionContext 中除了包含根对象、命名对象外,还包含了由这些 Struts2 标签所产生的对象。对于根对象的访问,可以直接通过 propertyName 的格式访问属性,其它非根对象的访问,必须使用指明对象所属的命名空间,在 Struts2 中,这个命名空间就是 ActionContext.getContext(),在 OGNL 中使用“#”来表示,例如: #request.name。

2 Struts2 OGNL 的使用

2.1 使用 OGNL 的场合

在 Struts2 中,可使用 OGNL 的场合有 Struts2 标签、Struts2 配置文件、属性文件及视图页面 Struts2 标签以外的地方。

在视图页面 Struts2 标签之外的地方使用语法是: \${ognl 表达式};在其它地方使用 OGNL 表达式时一般需要借助于操作符:“#”、“%”和“\$”,具体使用哪种操作符需要根据使用的场合来定。

1. “#”操作符的使用^[12]。

(1) 访问 OGNL 上下文非根对象。

在视图页面中访问 OGNL 上下文中的根对象,可直接通过 Struts2 标签来访问,访问方法是直接将根对象的属性名所表示的 OGNL 表达式赋给 Struts2 标签的相应属性,例如:<s:property value=“person.name”/>,该代码将查找 ValueStack 中的当前 action 实例的 person 属性的 name 值;如果需要访问 OGNL 上下文中

的非根对象,则需要使用 #Object. PropertyName 的格式,例如:使用 #session.userName 表达式来访问 OGNL 上下文的 session 对象中的 userName 属性。

(2) 投影和选择集合。

(3) 构造 Map 对象。

2. “%”操作符的使用。

计算 OGNL 表达式的值时需要使用 % 来操作 OGNL 表达式,格式是 % {ognl},例如:<s:label name=“name” label=“% {#session.username}”/>,该代码将通过 % 来获取 session 对象中 username 的属性值进而赋给 label 属性。

3. “\$”操作符的使用。

“\$”操作符主要用于在属性文件和 Struts2 配置文件中引用 OGNL 表达式。

在属性文件中引用 OGNL 表达式,可以获取属性文件中某配置项内已定义了的属性值或 Action 的属性值,例如在数据有效性校验文件中,对某个字段的取值范围指定了 min 和 max 范围,则在 <message> 中可使用 \${min} 和 \${max} 来引用取值范围。

Struts2 配置文件中引用 OGNL 表达式,可以获取 Action 的属性值,例如在当前 Action 实例中存在 category 属性,则可在配置文件中使用 <result>admin.jsp? type= \${category} </result>。

2.2 进行数据类型转换

在 Struts2 中,封装了基于 OGNL 表达式的类型转换器,该转换器除了可以完成字符串与基本类型及其包装类之间的类型转换外,还可以将普通请求参数转换成复合类型对象。

2.3 数据在 View 层和 Controller 层之间的流转

1. 数据从 Controller 层向 View 层流转。

当视图页面存在 Struts2 标签时,Struts2 框架会根据标签的 value 或 name 等属性的 OGNL 表达式来调用控制器 Action 的相应属性的 Getter 方法,Getter 方法执行时首先在 ValueStack 中寻找该 Action 对象的相应属性,找到后将属性值返回给视图,从而完成数据从 Controller 层向 View 层的流转。

2. 数据从 View 层向 Controller 层流转。

对于每个 action 请求,Struts2 首先将建立一个与相应 Action 对应的 ActionContext 作为 OGNL 的上下文环境和 ValueStack,并且把 Action 压入 ValueStack;接着在请求进入 Action 代码前,通过 Struts2 的 params 拦截器,解析视图页面上传过来的 HTTP 参数,并调用对应 HTTP 参数的属性 Setter 方法对 ValueStack 中的 Action 进行设置,在设置过程中需调用 Struts2 的基于 OGNL 的内置类型转换器,以实现字符串类型与其它基本数据类型或复合类型的转换。

3 Struts2 OGNL 表达式语言的应用

下面以电子商务系统中的产品发布管理模块为例来说明 OGNL 的使用。假设有这样的需求:用户发布产品成功后,页面跳转到如表 1 所示的所发布产品类型的产品列表页面。

表 1 产品列表(产品类型:数码家电)

序号	产品名称	产品型号	价格	生产厂家	操作	
1	数码相机	DSC-WX7	12990.0	索尼	删除	更新
2	数码相机	IXUS230 HS	1699.0	佳能	删除	更新
3	苹果手机	iPhone 4S 32G	5699.0	苹果	删除	更新
4	三星手机	Galaxy Note I9220	5499.0	三星	删除	更新
5	飞利浦 MP4	SA028304K/93	299.0	飞利浦	删除	更新
6	卡西欧电子辞典	E-B99 BU	2100.0	卡西欧	删除	更新

实现上述功能涉及了多个文件,其中包括有添加产品的 addProduct. jsp 页面、按产品类型列表产品的 listProduct. jsp 页面,还有执行业务逻辑控制的 ProductAction 动作类,以及实现业务逻辑的 ProductService 接口及其实现类 ProductServiceImpl 类和用于封装传输数据的 Product 类等多个文件。

下面将介绍其中涉及 OGNL 表达式语言的相关文件中的代码。

1. 发布产品页面。

使用 addProduct. jsp 页面实现产品的添加,该页面给用户提供了一个 Struts2 表单来发布产品信息,Struts2 表单代码如下:

```
<s:form action="addProduct" namespace="/product">
<s:textfield name="product.pname" label="产品名称"/>
<s:textfield name="product.model" label="产品型号"/>
<s:textfield name="product.price" label="产品价格"/>
<s:textfield name="product.category" label="产品类型"/>
<s:textfield name="product.manufacturer" label="生产厂家"/>
<s:textarea name="product.intro" label="产品简介"/>
<s:submit value="添加产品"/>
</s:form>
```

action="addProduct" 说明了表单中的数据需要提交给 addProduct. action 处理,另外,用于发布产品信息的每个表单域的 name 属性都使用了 OGNL 表达式,该表达式实现了将用户填写的产品信息封装到名称为 product 的对象中,其中 product 对象是一个 VO,需要在 addProduct. action 对应的 ProductAction 动作类中声明。

2. 封装产品信息类。

在该模块中使用了 Product 类来封装产品信息,代码如下:

```
public class Product {
private Integer id;
```

```
private String pname;
private String model;
private float price;
private String manufacturer;
private String category;
private String intro;
//省略 Setter 和 Getter 方法
}
```

3. 业务逻辑控制类。

使用 ProductAction 类来实现业务逻辑的控制,代码如下:

```
public class ProductAction extends ActionSupport {
private Integer id;
private String category;
private String cname;
private Product product;
private List<Product> products;
private ProductService productService;
Map session = ActionContext.getContext().getSession();
public ProductAction() {
productService = new ProductServiceImpl();
}
//省略 Setter 和 Getter 方法
public String addProduct() throws Exception {
cname = URLEncoder.encode(product.getCategory(), "utf-8");
productService.addProduct(product);
return SUCCESS;
}
public String listProduct() throws Exception {
products = productService.findAllProductByCategory(category);
session.put("category", category);
return SUCCESS;
}
public String delProduct() throws Exception {
category = (String) session.get("category");
cname = URLEncoder.encode(category, "utf-8");
productService.delteProduct(id);
return SUCCESS;
}
//省略其它业务方法
}
```

在 ProductAction 中声明了一个用来封装客户端参数的属性变量 product,其类型为 Product 类,通过 product 属性和 addProduct. jsp 页面中的 OGNL 表达式,addProduct. action 在执行 addProduct() 业务之前,首先通过 Struts2 的内置拦截器 params 调用 setProduct() 方法将客户端提交的产品信息转换成了 product 复合对象,在执行 setProduct() 的过程中,通过执行 Product 类中

的相应属性的 Setter 方法来完成 product 对象的设置操作。

为了在成功发布产品后列出与所发布产品类型下的所有产品,需要对另一个动作发起请求,这里设定该动作为 listProduct. action。由于在列出产品时需要知道所发布的产品类型,所以在对 listProduct. action 发出请求时必须附上查询参数值为产品类型,该请求 URL 的设置 struts.xml 中进行。

另外,为了在列表产品页面中显示产品类型以及删除某条产品信息后能返回到该列表产品页面,需要将产品类型数据在页面之间进行传递,这可通过在 listProduct() 方法中将查询参数传过来的产品类型保存到 session 对象的 category 属性中来实现,这样在列表产品页面就可以使用 OGNL 表达式来访问 session 对象的属性值了,而在 deleteProduct() 方法中,则可以通过 session 对象的 get() 方法来获取属性值,从而可将所获取的属性值作为查询参数使用。

4. 列表产品页面。

使用 listProduct. jsp 页面列出所发布的产品类型下的所有产品,代码如下:

```
<div style="height:28">产品列表 &gt;&gt;&gt; 产品类型:
<s:property value="#session.category"/></div>
.....
<s:iterator value="products" status="st" var="p">
<tr align="center" height="28">
<td><s:property value="#st.count"/></td>
<td><s:property value="#p.pname"/></td>
<td><s:property value="#p.model"/></td>
<td><s:property value="#p.price"/></td>
<td><s:property value="#p.manufacturer"/></td>
<td><s:a href="product/delProduct.action? id=% {#p.id}"
onclick="return del()">删除</s:a></td>
<td><s:a href="product/updateProduct.action? id=% {#p.id}">更新</s:a></td>
</tr>
</s:iterator>
```

表 1 上面显示的产品类型是通过<s:property value="#session.category"/>来获取的,标签的 value 属性值是一个 OGNL 表达式,用于访问 session 非根对象的 category 属性值。表格中的每个产品信息,则通过 s:iterator 标签迭代输出。

s:iterator 标签的 status 属性用于指定迭代时产生的 IteratorStatus 实例,以便通过它来产生表格中的序号;标签中的 var 属性则将迭代时生成的对象放入 Stack Context,因而访问时必须要在对象名称前面使用“#”。每次迭代产生的对象被命名为 p,所以需要使

用上述代码中,还使用了 OGNL 表达式的操作符 % 来计算超链接 URL 的查询参数 id 的值。

5. struts.xml 配置文件。

在 struts.xml 配置文件中配置了多个逻辑动作,分别用来执行产品的发布、删除和列表等操作,代码如下:

```
<action name="addProduct" class="action.ProductAction"
method="addProduct">
<result type="redirect">listProduct.action? category = $
{cname}</result>
</action>
<action name="listProduct" class="action.ProductAction"
method="listProduct">
<result name="success">/listProduct.jsp</result>
</action>
<action name="delProduct" class="action.ProductAction"
method="delProduct">
<result type="redirect">listProduct.action? category = $
{cname}</result>
</action>
```

添加产品和删除产品的动作成功执行后,都将请求重定向到 listProduct. action,在重定向请求的 URL 都附上了查询参数,该参数值通过 OGNL 表达式的操作符 \$ 来获取 ProductAction 类实例的 cname 属性值。

4 结束语

OGNL 是 Struts2 默认的表达式语言,相比于其他表达式语言,OGNL 提供了更强大的功能,这主要体现在它解决了数据从 View 层传递到 Controller 层时的不匹配性,实现数据类型的自动转换,可以方便地访问 OGNL 上下文中对象的任何属性、方法以及可以方便的操作集合及对象等优点。Struts2 框架主要用于实现 Web 应用体系结构中的表现层,而 OGNL 的使用可以极大地提高 Struts2 框架的表现能力,所以,OGNL 的掌握程度决定了能否灵活并充分使用 Struts2 所提供的功能。

参考文献:

- [1] 符光宝,邵定宏,李兰友. 基于 Struts 框架的档案管理系统应用研究[J]. 计算机工程与设计,2008,29(18):2100-2105.
- [2] 刘婷婷,蒋玉明,傅静涛. 集成 J2EE 框架构建可复用的电子商务系统[J]. 计算机工程,2008,26(11):200-202.
- [3] 闫术卓,杨 强. Struts2 技术详解:基于 WebWork 核心的 MVC 开发与实践[M]. 北京:电子工业出版社,2008.
- [4] 刘 斌,王 最. Struts, Spring, Hibernate 框架在 OA 开发中的应用[J]. 计算机技术与发展,2010,20(1):151-154.

(下转第 190 页)

条件为:

If(ammo. $z \leq 0$ 即弹丸世界坐标的 z 分量为小于或等于零时)

2.3 仿真实例

根据上述模型和算法编程实现对爆炸特效的仿真如图 5 所示。利用同样的步骤,分析炮口火焰和舰艇水幕特效的组成和运动方式^[11,12],得到仿真结果如图 6 所示。

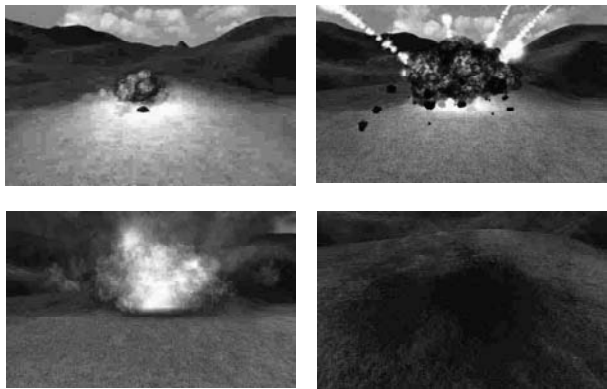


图 5 爆炸特效

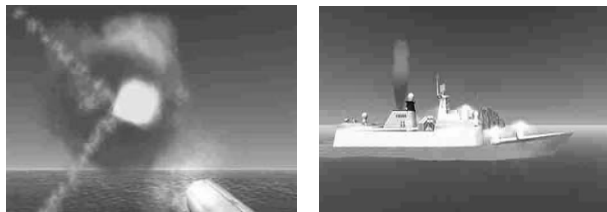


图 6 炮口火焰和舰艇开启水幕特效

文中对战场特效的模拟使用 Torque T3D 引擎,编程基于 torque script 脚本语言完成。

3 结束语

战场特效模拟是虚拟战场视景仿真的重点,也是难点,逼真的特效能够增强视景系统的真实性。文章对用粒子系统模拟战场特效的过程进行了研究,针对其随机性和复合性的特点,提出了一种基于基础特效

库,利用通用模型对基础特效进行组合和配置的方法,可以实现对多种战场特效的模拟,省却了为每一种特效单独进行设计,提高了代码的可重用性。实验表明:所生成的特效完全满足了系统的实时性要求,整个模拟过程从视觉上显得比较逼真,能够有效提高虚拟场景的沉浸感。

参考文献:

- [1] 范乃梅,马军霞. 基于粒子系统的烟雾和爆炸模拟[J]. 郑州轻工业学院学报,2009,24(5):77-80.
- [2] 余 洋. 基于粒子系统的爆炸效果模拟[D]. 合肥:安徽大学,2010.
- [3] William T R. Particle system—a technique for modeling a class of fuzzy objects[J]. Computer Graphics,1983,17(3):359-376.
- [4] 王子才. 仿真技术发展及应用[J]. 中国工程科学,2003,5(2):40-44.
- [5] 詹荣开,罗世彬,贺汉根. 用粒子系统理论模拟虚拟场景中的火焰和爆炸过程[J]. 计算机工程与应用,2001(5):91-92.
- [6] 张 芹,吴慧中,谢隽毅,等. 基于粒子系统的火焰模型及其生成方法研究[J]. 计算机辅助设计与图形学学报,2001,13(1):78-82.
- [7] Gross M E. A real time particle system for display of ship wakes[J]. IEEE Computer Graphics and Applications,1990,10(3):30-35.
- [8] 汪继文,余 洋,李玉梅. 基于粒子系统的陨石爆炸模拟[J]. 计算机技术与发展,2010,20(6):118-120.
- [9] 王红霞. 基于粒子系统的烟花仿真与实现[D]. 长春:东北师范大学,2008.
- [10] 郑 岌. 虚拟战场绘制的数据管理与特效生成[D]. 长沙:国防科学技术大学,2007.
- [11] Fosts N, Metaxas D. Realistic Animation of Liquid[J]. Graphical Models and Image Processing,1996,58(5):471-483.
- [12] 赵 欣,李凤霞,战守义. 基于粒子系统的舰船航迹仿真[J]. 计算机工程,2008,34(15):22-24.

(上接第 186 页)

- [5] 丁 波,晁爱农. 基于 Struts2 框架的 AJAX 开发研究[J]. 计算机工程与设计,2009,30(16):3910-3913.
- [6] Brown D. Struts2 in Action[M]. American:Manning Publications,2008.
- [7] 王 彬,张计龙. Struts2 框架的改进与实现[J]. 大连海事大学学报,2010,36(2):75-77.
- [8] Downpour. Struts2ognl[EB/OL]. 2011-06-01. <http://struts2.group.iteye.com/group/wiki/1353-ognl-catalyst-for-data-operation-in-struts2>.
- [9] Roughley I. Practical Apache Struts2 Web 2.0 Projects[M]. American:Apress,2007.
- [10] Qidaoxp. ValueStack 与 OGNL[EB/OL]. 2009-10-08. <http://qidaoxp.iteye.com/blog/482871>.
- [11] 李 刚. 轻量级 Java EE 企业级应用实战—Struts2+Spring3+Hibernate 整合开发[M]. 第 3 版. 北京:电子工业出版社,2011.
- [12] Kurniawan B. Struts2 Design and Programming: A Tutorial[M]. American:Brainysoftware,2008.

Struts2 OGNL表达式语言的研究与应用

作者：聂常红，唐远强，冯相忠

作者单位：聂常红(广州大学华软软件学院, 广东 广州510990)，唐远强(浙江海洋学院 数理与信息学
院, 浙江 舟山316000)，冯相忠(浙江海洋学院 机电工程学院, 浙江 舟山316000)

刊名：计算机技术与发展

英文刊名：Computer Technology and Development

年，卷(期)：2013(3)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_wjfz201303048.aspx